



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Μελέτη, σχεδίαση και ανάπτυξη λογισμικού διαχείρισης ιατρικών εξετάσεων. The study, design, and development of an application for the management of medical examinations.
Όνοματεπώνυμο Φοιτητή	ΕΝΡΙ ΒΑΓΓΕΛΗΣ
Πατρώνυμο	ΠΟΛΥΖΩΗΣ
Αριθμός Μητρώου	ΜΠΠΛ 17004
Επιβλέπων	ΑΛΕΠΗΣ ΕΥΘΥΜΙΟΣ , ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

Ημερομηνία Παράδοσης	Ιανουάριος 2021
----------------------	------------------------

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Αλέπης Ευθύμιος
Αναπληρωτής
Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Κων/νος Πατσάκης
Αναπληρωτής
Καθηγητής

Περιεχόμενα

Περίληψη	4
Abstract	4
Εισαγωγή	5
Ανασκόπηση πεδίου	6
Παρουσίαση και χρήση εφαρμογής	9
Έναρξη εφαρμογής. (τοπικά / localhost)	9
Οδηγίες χρήσης εφαρμογής μέσω του Postman	10
1. Διαχειριστές	11
2. Χρήστες	13
3. Εξετάσεις	17
Παρουσίαση εφαρμογής μέσω προγράμματος περιηγητή (Browser)	20
Login	20
Σελίδα εξετάσεων	21
Σελίδα προσθήκης εξέτασης	22
Προβολή Λογαριασμού	22
Αρχιτεκτονική συστήματος	23
Βάση δεδομένων Mongo DB	23
Backend node JS	25
Frontend React JS	31
Συμπεράσματα και μελλοντικές επεκτάσεις	32
Βιβλιογραφία	33

Περίληψη

Στόχος της παρούσας διπλωματικής διατριβής είναι η μελέτη, ο σχεδιασμός και η ανάπτυξη μιας σύγχρονης ιατρικής εφαρμογής. Η εφαρμογή αποσκοπεί στην καλύτερη διαχείρισή ιατρικών εξετάσεων από του ασθενείς και τους ιατρούς. Για την υλοποίηση της εφαρμογής My Health απαιτήθηκε η κατασκευή της αυτής καθ' αυτής εφαρμογής του ασθενή / ιατρού, μιας διαδικτυακής πλατφόρμας και της βάσης δεδομένων.

Η εφαρμογή my Health θέλει να συγκεντρώσει σε ένα σύστημα όλες τις διαδικασίες που απαιτείται για την οργάνωση των εξετάσεων ενός ασθενή αλλά και την ταυτόχρονη παρακολούθηση τους από τον επιβλέποντα ιατρό. Επομένως, ο ασθενής θα γνωρίζει ποτέ εξεταστικές και ποιες ήταν οι παρατηρήσεις του ιατρού, θα μπορεί ακόμα και να προσθέτει / διαγραφεί / ολοκληρώνει εξετάσεις. Επίσης, ο ιατρός θα μπορεί να οργανώσει τους ασθενείς του, να τους επιβλέπει μέσω των εξεταστών, να κρατάει ιστορικό και να διαχειρίζεται τις εξετάσεις τους.

Για την υλοποίηση της παραπάνω εφαρμογής χρησιμοποιήθηκαν σύγχρονες ανοιχτού κώδικα τεχνολογίες. Για την διεπαφή του χρήστη επιλέχθηκε η γλώσσα JavaScript μέσω της βιβλιοθήκης React, για backend το framework NodeJS και για βάση δεδομένων η μη σχεσιακή βάση mongo DB.

Abstract

The purpose of this thesis is the study, research, and development of telematics web applications. The application aims to manage medical examinations by patient better. The application provides instant communication between doctors and patients at the same time.

My Health app is trying to collect all the procedures in one application to organize users' examinations. Moreover, the patient ll be able to know his medical history with dates and comments. Also, he can arrange new inspections and manage them. Doctors ll be able to organize their patients properly, supervise them, create a medical history, and finally execute them.

To meet the current trends, it was essential to use open source libraries and frameworks such as react JS, NodeJS, and Mondo DB.



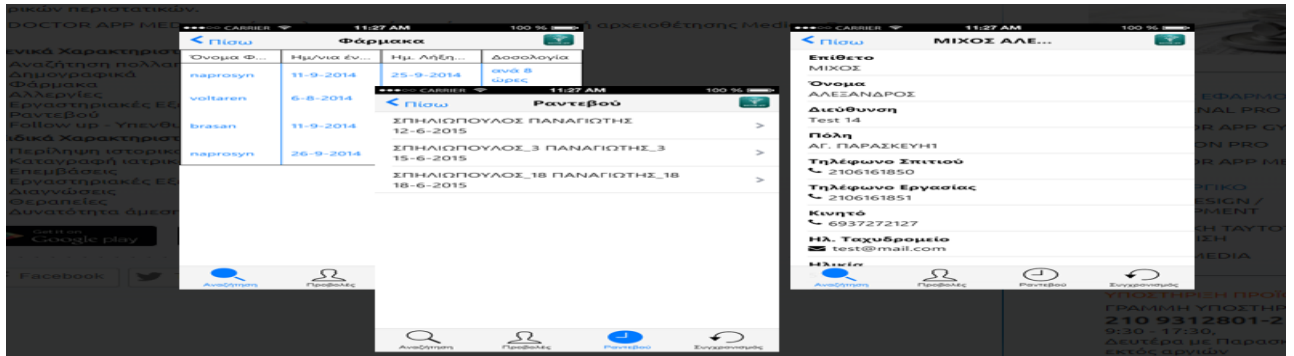
Εισαγωγή

Στην σημερινή εποχή ο χώρος της υγείας αποτελεί ένα πεδίο έρευνας και εφαρμογής τεχνολογικών επιτευγμάτων τόσο στον ιδιωτικό τομέα όσο και στο δημόσιο. Παρατηρείται ότι με τη ραγδαία ανάπτυξη της πληροφορικής επωφελείται σε σημαντικό βαθμό και η ιατρική. Με την ανάπτυξη των δυο αυτών επιστημών επωφελούνται οι ίδιοι οι ασθενείς καθώς οι διαγνώσεις είναι πιο ακριβείς, τα νοσοκομεία πιο οργανωμένα και ασθενείς κερδίζουν χρόνο και χρήμα. Στόχος της εφαρμογής my-Health είναι βοηθήσει του ιατρούς και ασθενείς να κερδίσουν χρόνο, να μειωθούν οι αποστάσεις μετακίνησής και να μειωθούν τα έξοδα τους. Όσον αφορά τους ιατρούς, η εφαρμογής τους βοηθάει στο να έχουν καλύτερη οργάνωση, να διαχειριστούν τους ασθενείς τους και να έχουν ανά πάσα στιγμή εικόνα για την κατάσταση των ασθενών τους. Από την άλλη μεριά οι ασθενείς μπορούν να διαχειριστούν τις εξετάσεις τους και να έχουν ένα ιατρικό ιστορικό.

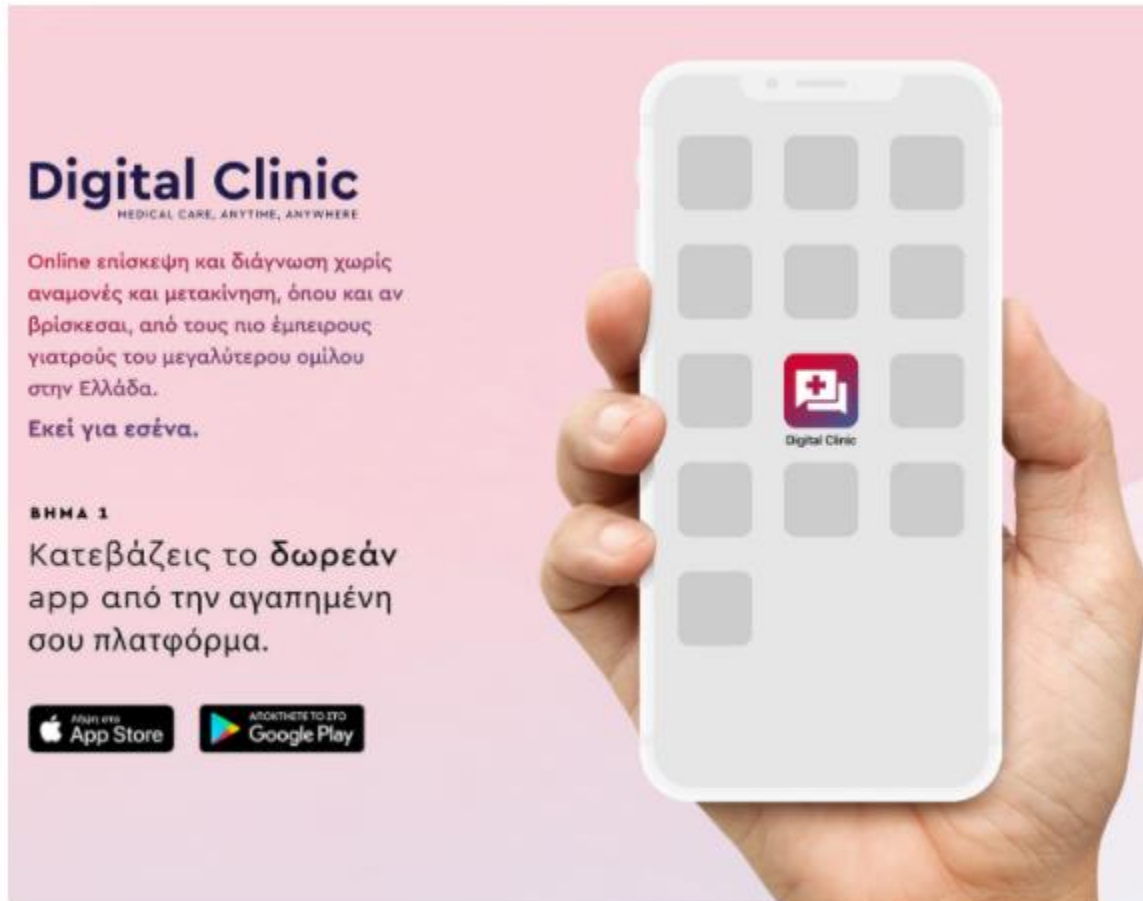
Η εφαρμογή my-Health δίνει τη δυνατότητα σε ασθενείς και ιατρούς να δημιουργήσουν τον δικό τους λογαριασμό. Ο ασθενής (user) μπορεί δημιουργήσει, να διαγράψει, να τροποποιήσει και να διαβάσει εξετάσεις, επίσης μπορεί να ανεβάσει αρχεία για τυχόν φυσικές εξετάσεις. Ο ιατρός (admin) από την άλλη μεριά μπορεί να δημιουργήσει, διαγράψει και τροποποιήσει χρήστες, επίσης έχει δικαιώματα και ως προς τις εξετάσεις των ασθενών του. Πιο αναλυτικά, με την εγγραφή τους στην εφαρμογή τους αποστέλλεται ένα email καλωσορίσματος. Ο ασθενής μετά την επιτυχημένη του είσοδο μπορεί να δει την κατάσταση των εξετάσεων του και με ένα click να τις τροποποιήσει.

Ανασκόπηση πεδίου

Η εφαρμογή Doctor App Med είναι διαθέσιμη σε IOS και android και προσφέρει τη δυνατότητα άμεσης πρόσβασης και επισκόπησης σε ραντεβού και εργασιακές εξετάσεις και πλήθος δεδομένων για την παρακολούθηση ιατρικών περιστατικών.



Η εφαρμογή “Digital Clinic” απευθύνεται σε ασθενείς που βρίσκονται μακριά από νοσοκομείο ή ιατρό, αντιμετωπίζουν ένα απλό θέμα υγείας ή δεν έχουν τη δυνατότητα και το χρόνο για να κλείσουν ραντεβού. Χωρίς αναμονή και χωρίς μετακίνηση, το app προσφέρει στον ασθενή τη δυνατότητα να λάβει έγκυρη ιατρική εκτίμηση και καθοδήγηση από παθολόγο του Ομίλου ΗΗG, μέσω διαλόγου (chat) και αποστολής φωτογραφιών ή αρχείων. Ο διάλογος αυτός καταλήγει σε διάγνωση, συνταγογράφηση φαρμακευτικής αγωγής ή, στις περιπτώσεις που κρίνεται απαραίτητο, παραπομπή στον κατάλληλο ιατρό και στις κατάλληλες ιατρικές εξετάσεις.



Digital Clinic
MEDICAL CARE, ANYTIME, ANYWHERE

Online επίσκεψη και διάγνωση χωρίς αναμονές και μετακίνηση, όπου και αν βρίσκεσαι, από τους πιο έμπειρους γιατρούς του μεγαλύτερου ομίλου στην Ελλάδα.

Εκεί για εσένα.

ΒΗΜΑ 1
Κατεβάζεις το δωρεάν app από την αγαπημένη σου πλατφόρμα.

ΚΑΤΕΒΑΣΤΕ ΤΟ ΑΠΟ ΤΗΝ
App Store

ΑΠΟΚΤΗΤΕ ΤΟ ΣΤΟ
Google Play

Η εφαρμογή **ΙΑΤΡΟΠΟΛΙΣ Medical Reports** δίνει την δυνατότητα παρακολουθήσεις αποτελεσμάτων ιατρικών.



ΙΑΤΡΟΠΟΛΙΣ Medical Reports

Infomed C.S Ιατρικό

★★★★★ 29

Ε Κατάλληλο για όλους

⚠ Δεν έχετε καμία συσκευή.

➕ Προσθήκη στη λίστα επιθυμιών

Εγκατάσταση

Εξέταση	Αποτέλεσμα	Ιστορικό	Μονάδες	Τιμές αναφοράς	Σχόλια
Creatinine (bl)	0.67	Προβολή...	mg/dL	0.70 - 1.20	
SGOT (AST)	14.1	Προβολή...	U/L	< 38.0	
SGPT (ALT)	21.0	Προβολή...	U/L	< 41.0	
GGT	83.0	Προβολή...	U/L	8.0 - 61.0	
ALP - Alkaline Phosphatase	94.0	Προβολή...	U/L	40.0 - 130.0	
Bilirubin Total	0.58	Προβολή...	mg/dL	0.00 - 1.00	
Bilirubin Conjugated (Bc)	0.24	Προβολή...	mg/dL	0.00 - 0.20	
Albumin	36.1	Προβολή...	g/L	35.0 - 52.0	
LDH	248.0	Προβολή...	U/L	135.0 - 225.0	
Calcium (Ca)	8.60	Προβολή...	mg/dL	8.60 - 10.20	
Magnesium	1.70	Προβολή...	mg/dL	1.60 - 2.40	
Sodium (Na)	140.0	Προβολή...	mmol/L	136.0 - 145.0	
Potassium (K)	3.60	Προβολή...	mmol/L	3.50 - 5.10	
Methotrexate	0.020	Προβολή...	mmol/l		

Ημ/νία ώρα
10/11/10 9:15:00 π.μ.
8/11/10 11:46:00 π.μ.
7/11/10 10:02:00 π.μ.
4/11/10 11:17:00 π.μ.
1/11/10 1:30:00 μ.μ.
29/10/10 11:13:00 π.μ.
28/10/10 7:42:00 μ.μ.
27/10/10 6:57:00 μ.μ.
25/10/10 6:44:00 μ.μ.
23/10/10 8:22:00 μ.μ.
4/10/10 9:35:00 π.μ.
29/9/10 2:01:00 μ.μ.
21/9/10 8:55:00 μ.μ.

Παρουσίαση και χρήση εφαρμογής

Έναρξη εφαρμογής. (τοπικά / localhost)

Από το τερματικό πληκτρολογούμε `npm start` (frontend) και `npm run server` (backend) ώστε να τρέξει η εφαρμογή. Το UI της εφαρμογής είναι διαθέσιμη στο URL `localhost : 3000`.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\avangeli\Desktop\playground\thesis-health-app> npm start
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

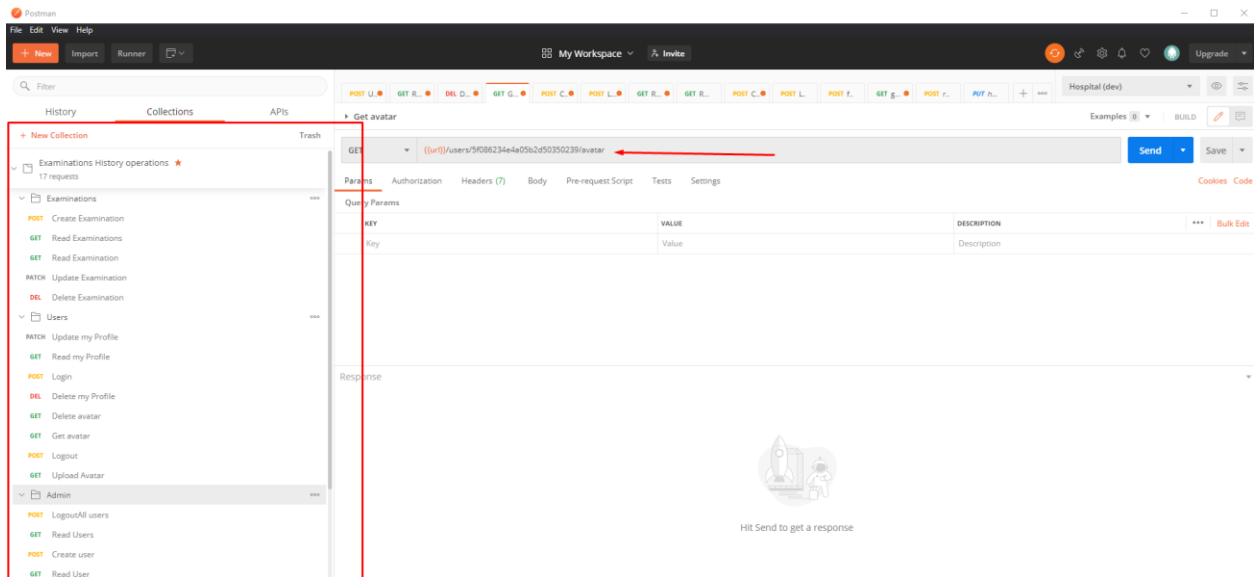
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\avangeli\Desktop\playground\thesis-health-app> npm run start:server
> my-health@0.1.0 start:server C:\Users\avangeli\Desktop\playground\thesis-health-app
> nodemon server/index.js

[nodemon] 2.0.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server/index.js`
(node:10660) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Server is up on port: 3000
```

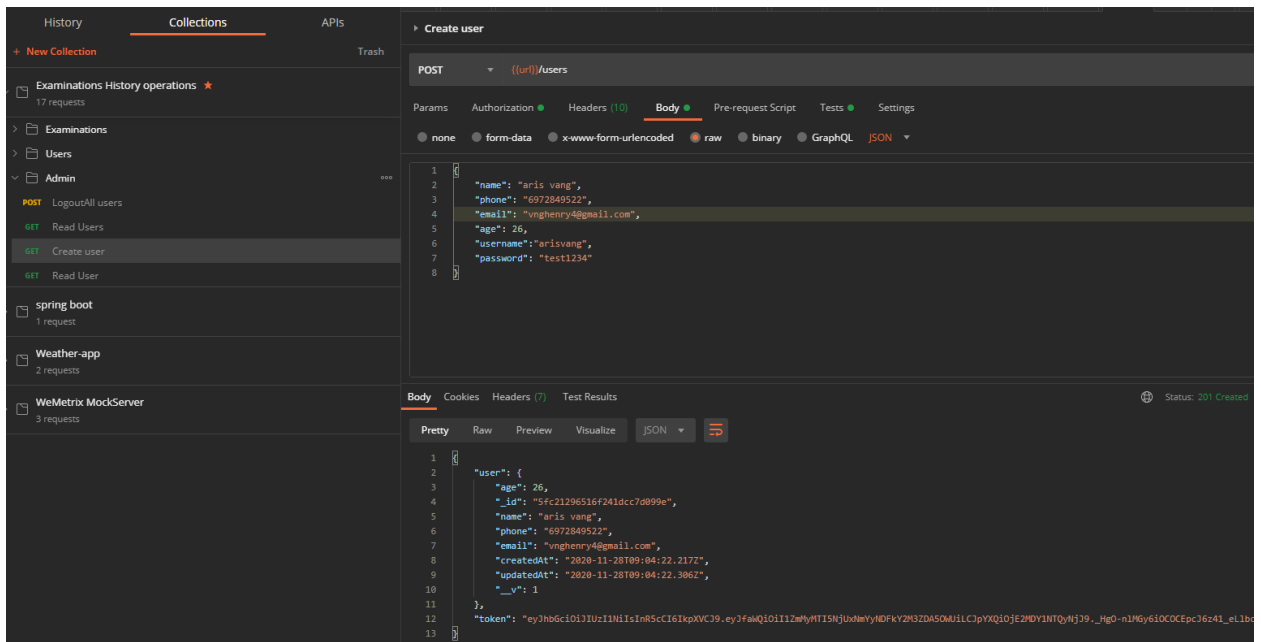
Οδηγίες χρήσης εφαρμογής μέσω του Postman

Για τον έλεγχο των APIs της εφαρμογής my-health χρησιμοποιείται η εφαρμογή postman. Το postman θεωρείται μια από τις πιο γνωστές και αναπτυγμένες εφαρμογές η οποία δίνει τη δυνατότητα στους προγραμματιστές να δοκιμάζουν τις υλοποιήσεις τους μέσω γραφικού περιβάλλοντος.



1. Διαχειριστές

Για τη δημιουργία χρήστη επιλέγουμε αριστερά Admin και στη συνέχεια create user. Ως body στέλνουμε τα στοιχεία του χρήστη και ο server μας απαντάει με τα στοιχεία καταχώρισης του χρήστη στη βάση μαζί με το authentication token. Ο διαχειριστής έχει πρόσβαση σε όλους χρήστες της εφαρμογής, μπορεί να τους διαχειριστεί, επίσης να επεξεργαστεί όλες τις εξετάσεις. Με κάθε δημιουργία χρήστη στέλνετε ένα email καλωσορίσματος σε αυτό αυτό που έχει δηλώσει κατά την εγγραφή του.



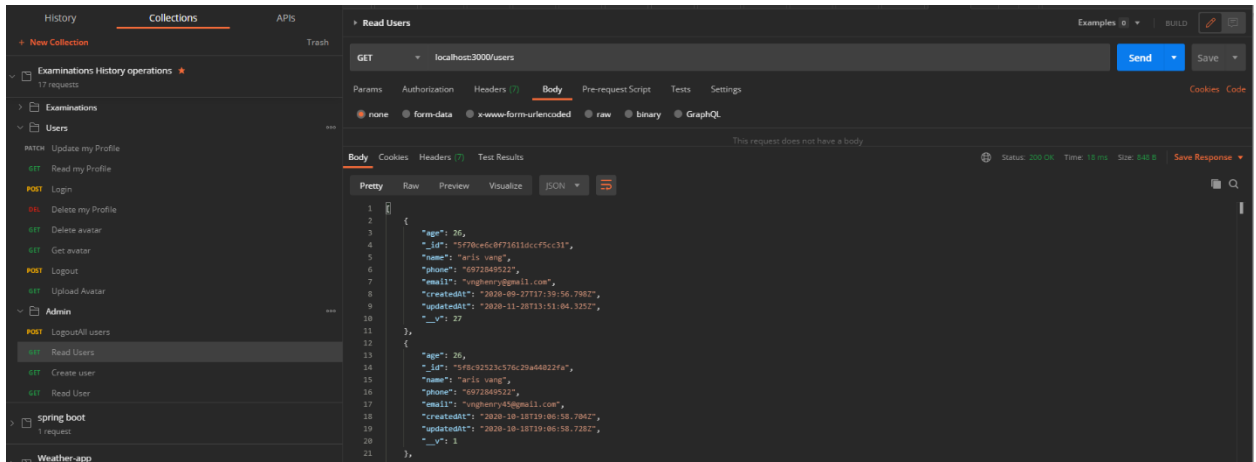
The screenshot shows the Postman interface with a POST request to `((url))/users`. The request body is a JSON object:

```
1 {
2   "name": "aris vang",
3   "phone": "6972849522",
4   "email": "vnghenry4@gmail.com",
5   "age": 26,
6   "username": "arisvang",
7   "password": "test1234"
8 }
```

The response body is a JSON object:

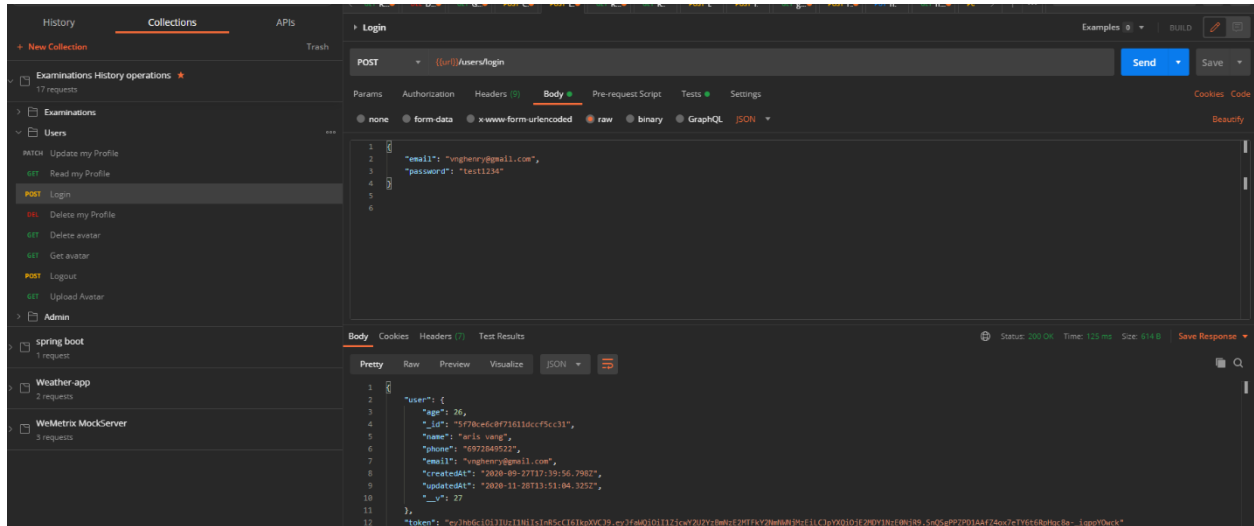
```
1 {
2   "user": {
3     "age": 26,
4     "_id": "5fc21296516f241dcc7d899e",
5     "name": "aris vang",
6     "phone": "6972849522",
7     "email": "vnghenry4@gmail.com",
8     "createdAt": "2020-11-28T09:04:22.217Z",
9     "updatedAt": "2020-11-28T09:04:22.386Z",
10    "_v": 1
11  },
12  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ3aWQiOiJ1Zm9udWVyc0FkY2h3ZDAsOmU1L0pvcXQ1OjE2PDIyMTQyMj9..._Hgo-nlM6y610C0CEpc36z41_el1b"
13 }
```

Η εφαρμογή δίνει τη δυνατότητα να υπάρχει και διαχειριστής ο οποίος μπορεί να διαβάσει όλους χρήστες να τους διαχειριστεί και να τους αποσυνδέσει από την εφαρμογή. Μέσω της μεθόδου GET και στο path `localhost:3000/users` η εφαρμογή επιστρέφει ένα json με πίνακα ο οποίος περιλαμβάνει όλους τους χρήστες.



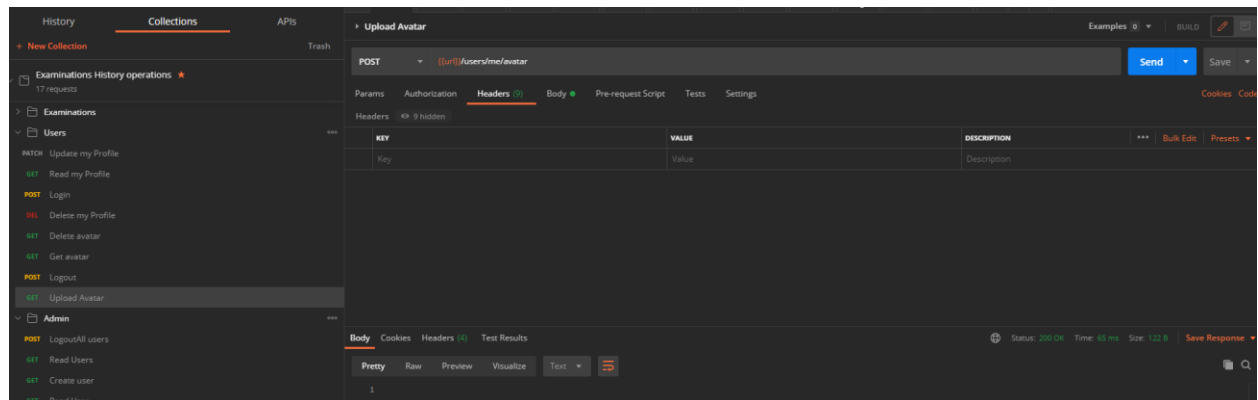
2. Χρήστες

Ο χρήστης προκειμένου να χρησιμοποιήσει τις υπηρεσίες της εφαρμογής θα πρέπει να κάνει είσοδο με το username και το password του. Με την είσοδό του ο χρήστης αποκτάει και το authentication token το οποίο πιστοποιεί την αυθεντικότητα του σε κάθε επικοινωνία με το backend. Στο body request και μέσω της μεθόδου POST και στο path localhost:3000/users/login η εφαρμογή επιστρέφει ένα json με τα στοιχεία του χρήστη.

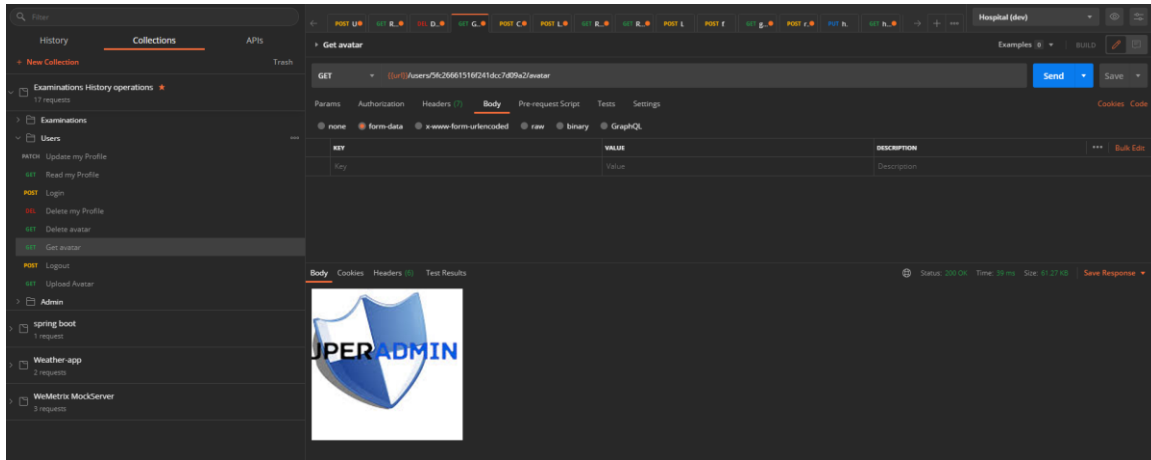


Διαχείριση φωτογραφίας λογαριασμού.

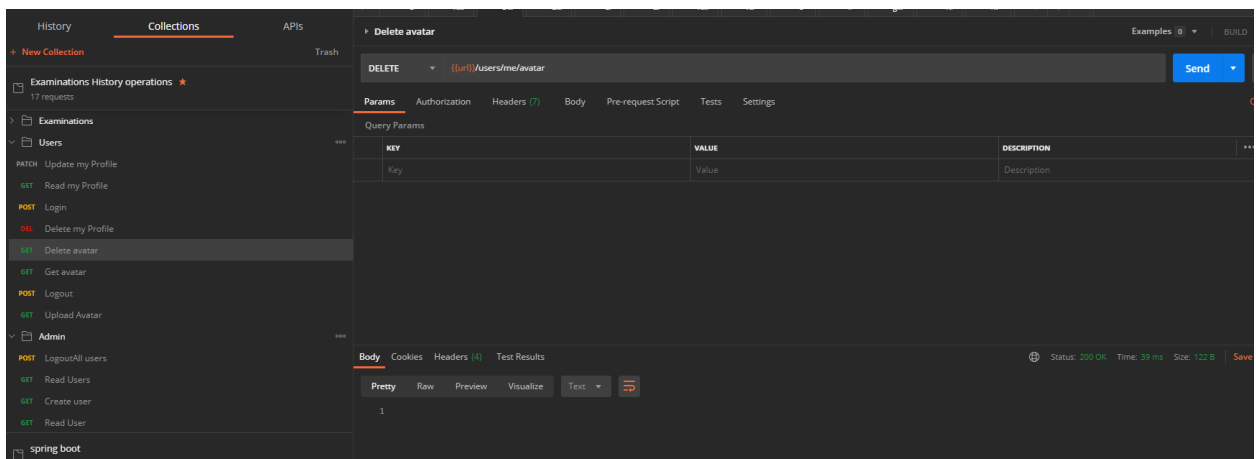
- i. Η εφαρμογή δίνει τη δυνατότητα στον χρήστη μέσω του προφίλ του να αναρτήσει τη δικιά του φωτογραφία (avatar). Μέσω της μεθόδου POST και στο path localhost:3000/users/me/avatar η εφαρμογή επιστρέφει ένα status OK.



ii. Εμφάνιση φωτογραφίας, όταν το service ολοκληρώνεται η φωτογραφία εμφανίζεται στους χρήστες. Μέσω της μεθόδου GET και στο path localhost:3000/users/:id/avatar η εφαρμογή επιστρέφει την φωτογραφία.

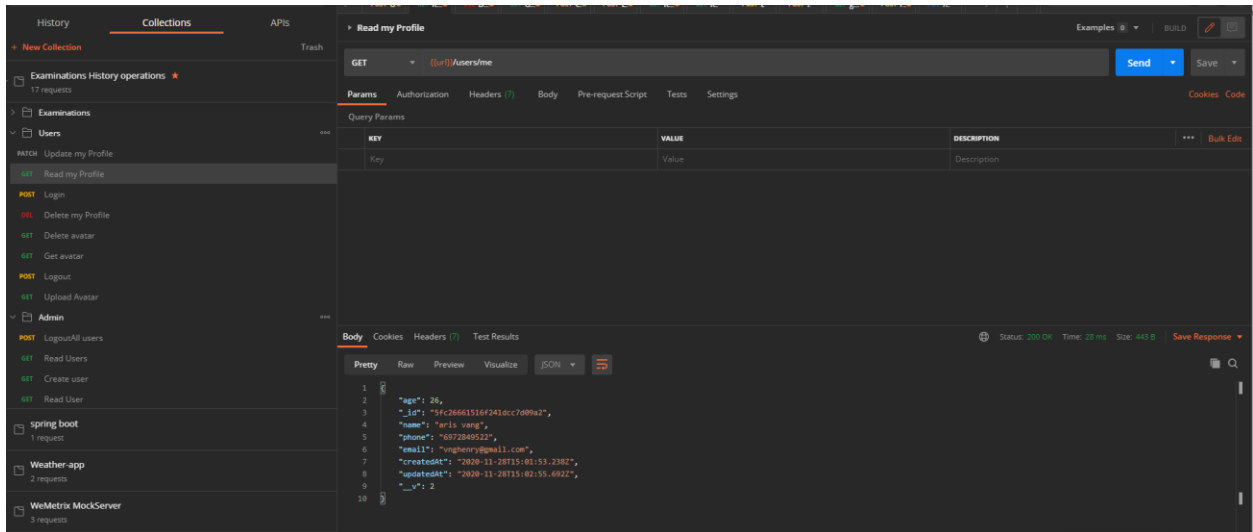


iii. Επίσης δίνεται η δυνατότητα στο χρήστη να διαγράψει την φωτογραφία του. Μέσω της μεθόδου DELETE και στο path localhost:3000/users/me/avatar η εφαρμογή διαγράφει την φωτογραφία.

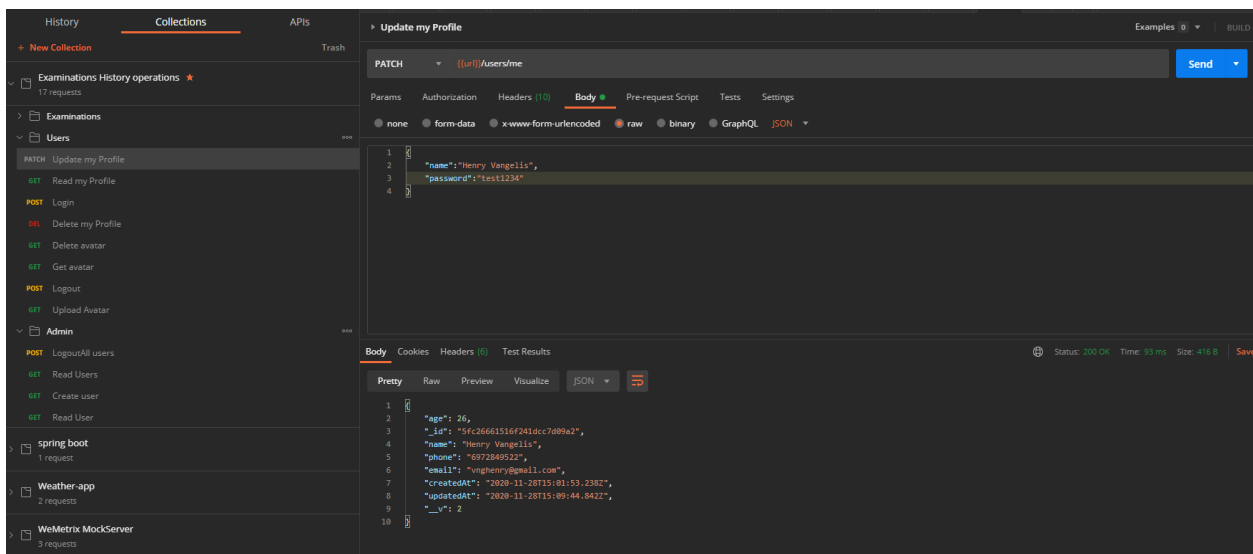


Διαχείριση Προσωπικού Λογαριασμού Χρήστη

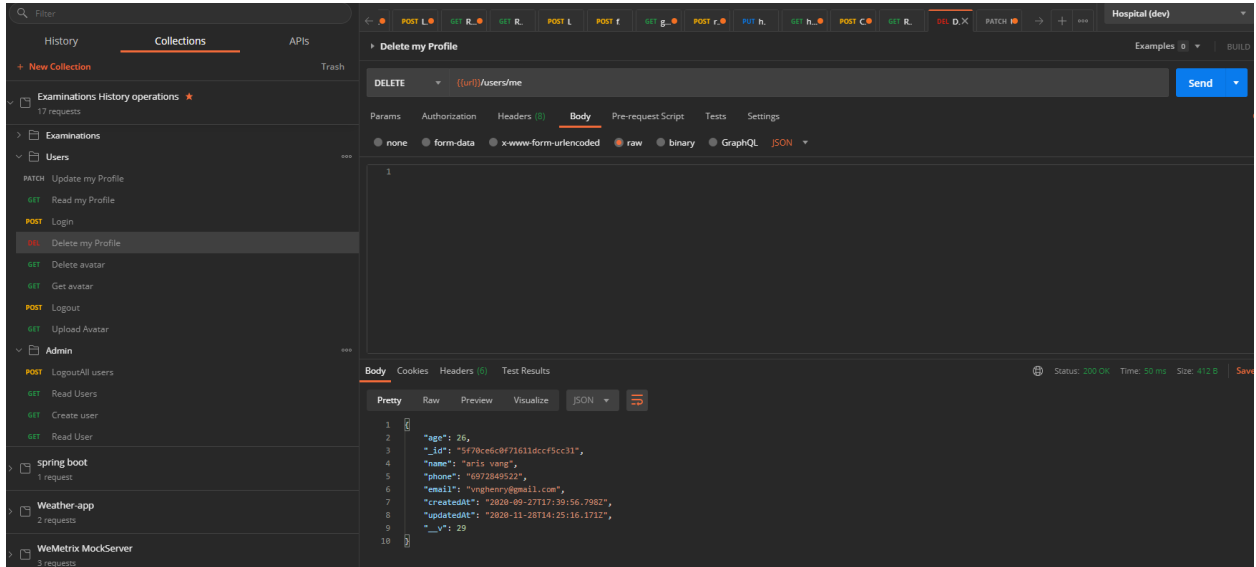
- i. Ο χρήστης μέσω ενός service θα μπορεί να βλέπει τα στοιχεία του. Μέσω της μεθόδου GET και στο path localhost:3000/users/me η εφαρμογή επιστρέφει ένα json με τα στοιχεία του χρήστη.



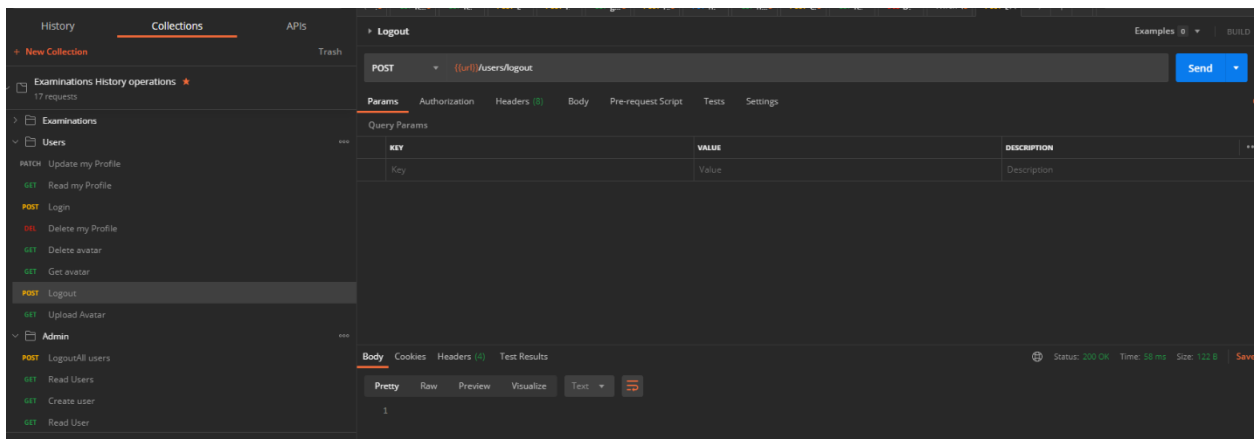
- ii. Επίσης θα μπορεί να στέλνει και τις αλλαγές του μετα την επεξεργασία του Λογαριασμού του. Μέσω της μεθόδου PATCH και στο path localhost:3000/users στέλνοντας ένα json με τις αλλαγές και η εφαρμογή επιστρέφει ένα json με τα καινούρια στοιχεία του χρήστη.



- iii. Ο χρήστης θα μπορεί να διαγράψει το λογαριασμό του. Μέσω της μεθόδου DELETE και στο path localhost:3000/users/me η εφαρμογή επιστρέφει ένα json με τα στοιχεία του χρήστη που διαγράφηκε.

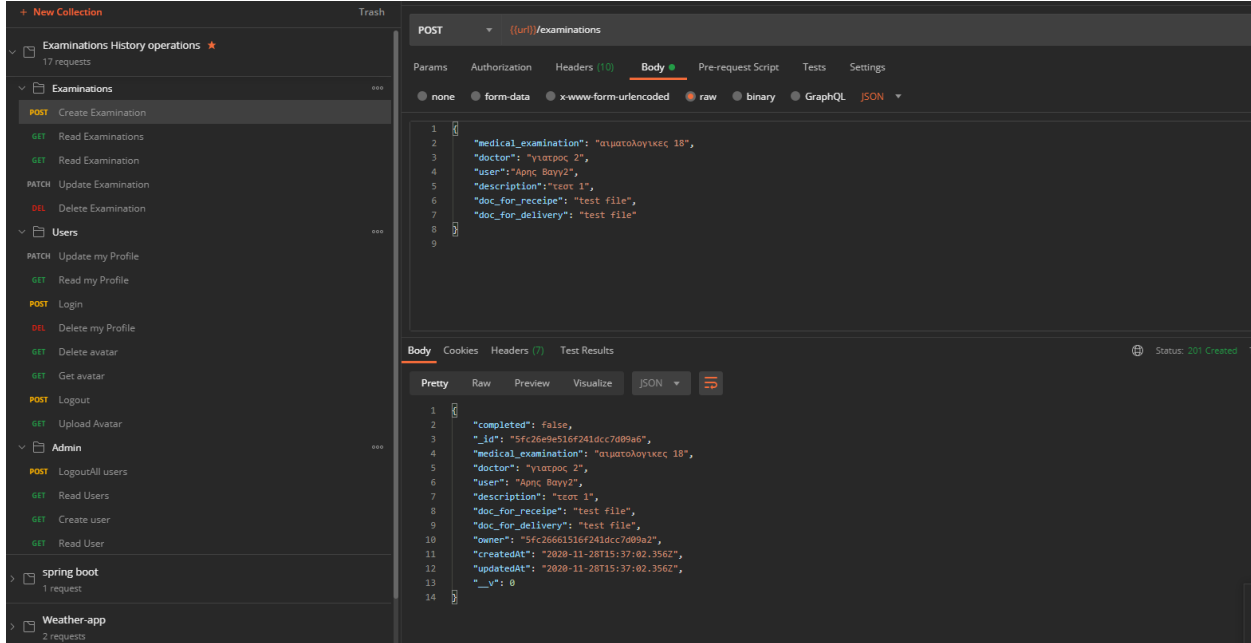


- iv. Τέλος να κάνει έξοδο από την εφαρμογή. Μέσω της μεθόδου POST και στο path localhost:3000/users/logout η εφαρμογή επιστρέφει status OK.

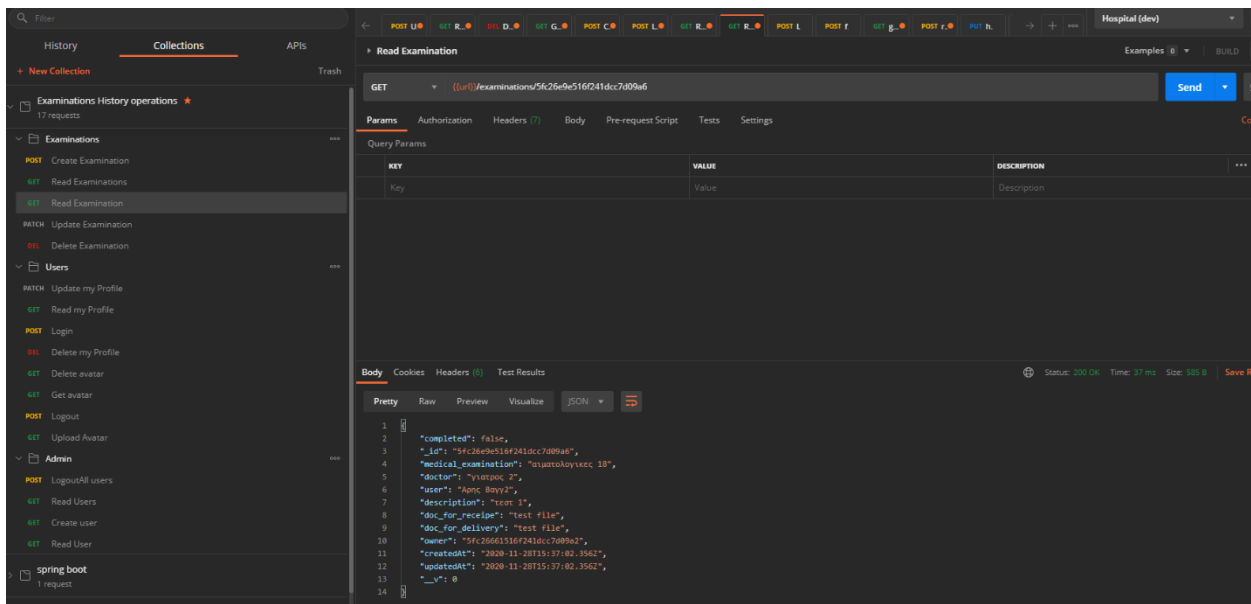


3. Εξετάσεις

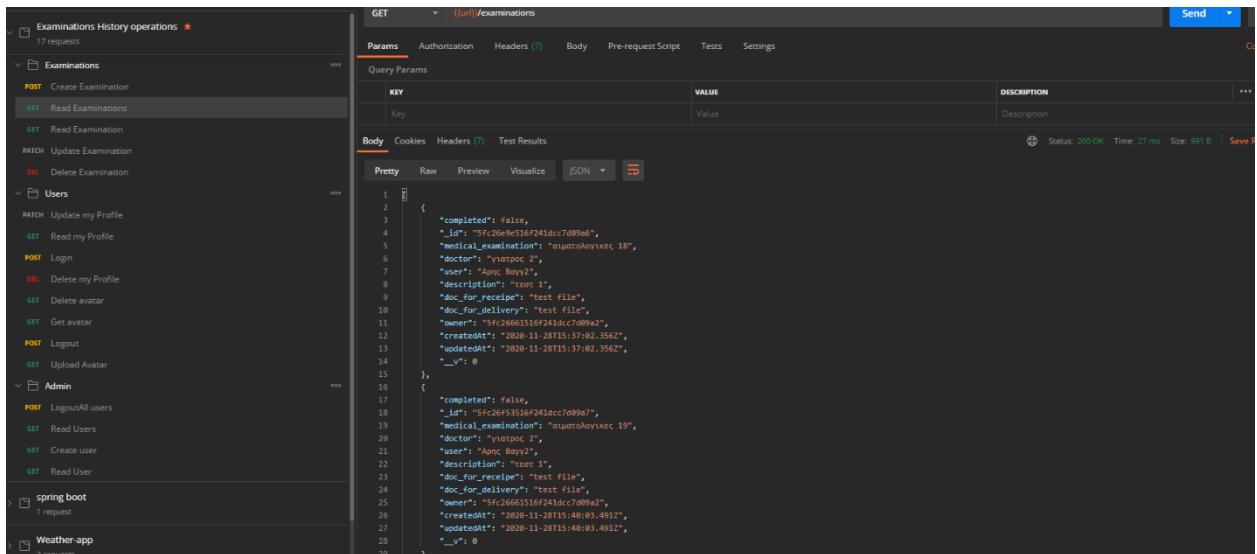
- i. Ο χρήστης μέσω της εφαρμογής μπορεί να δημιουργήσει τις δίκες του εξετάσεις. Μέσω της μεθόδου POST και στο path localhost:3000/examinations η εφαρμογή επιστρέφει ένα json με τα στοιχεία της καινούριας.



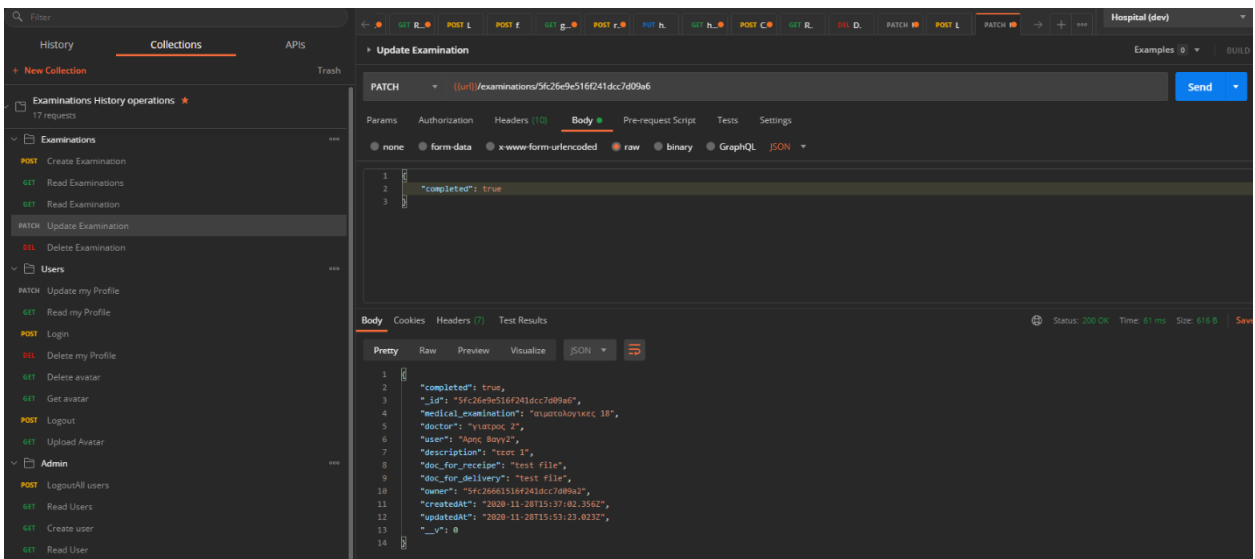
- ii. Ο χρήστης μέσω της εφαρμογής μπορεί να δει συγκεκριμένη δική του εξέταση. Μέσω της μεθόδου GET και στο path localhost:3000/examinations/:id η εφαρμογή επιστρέφει ένα json με τα στοιχεία της εξέτασης.



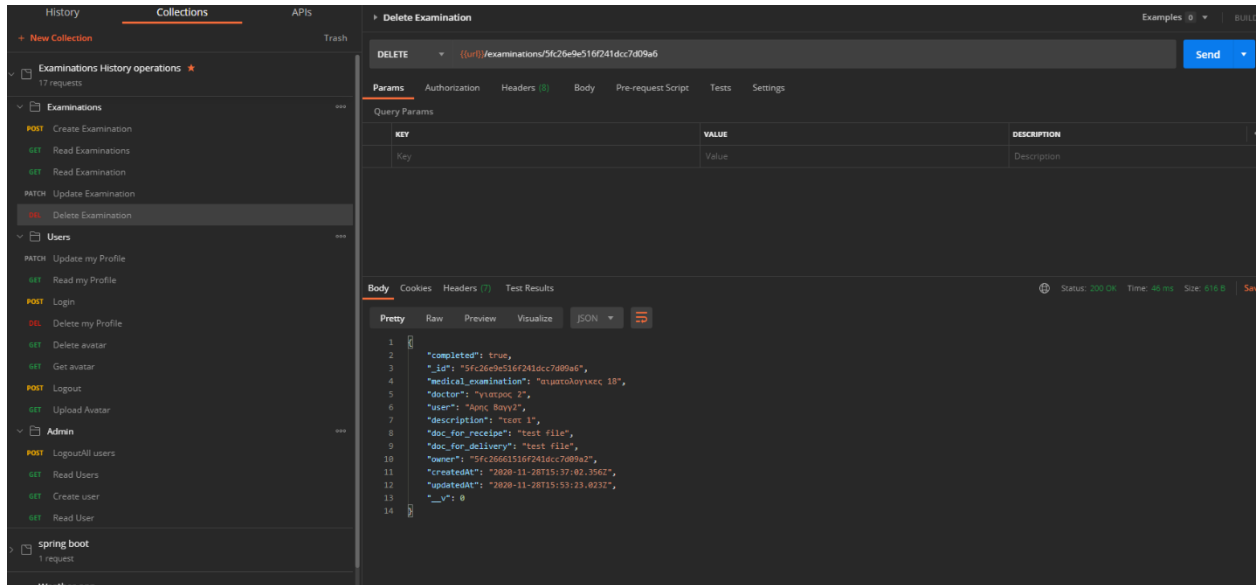
- i. Ο χρήστης μέσω της εφαρμογής μπορεί να δει τις δίκες του εξετάσεις. Μέσω της μεθόδου GET και στο path localhost:3000/examinations η εφαρμογή επιστρέφει ένα json με πίνακα που περιλαμβάνει όλες τις εξετάσεις.



- iii. Ο χρήστης μέσω της εφαρμογής μπορεί να επεξεργαστεί τις εξετάσεις του. Μέσω της μεθόδου PATCH και στο path localhost:3000/examinations/:id η εφαρμογή επιστρέφει ένα json με τα στοιχεία της τροποποιημένης εξέτασης.



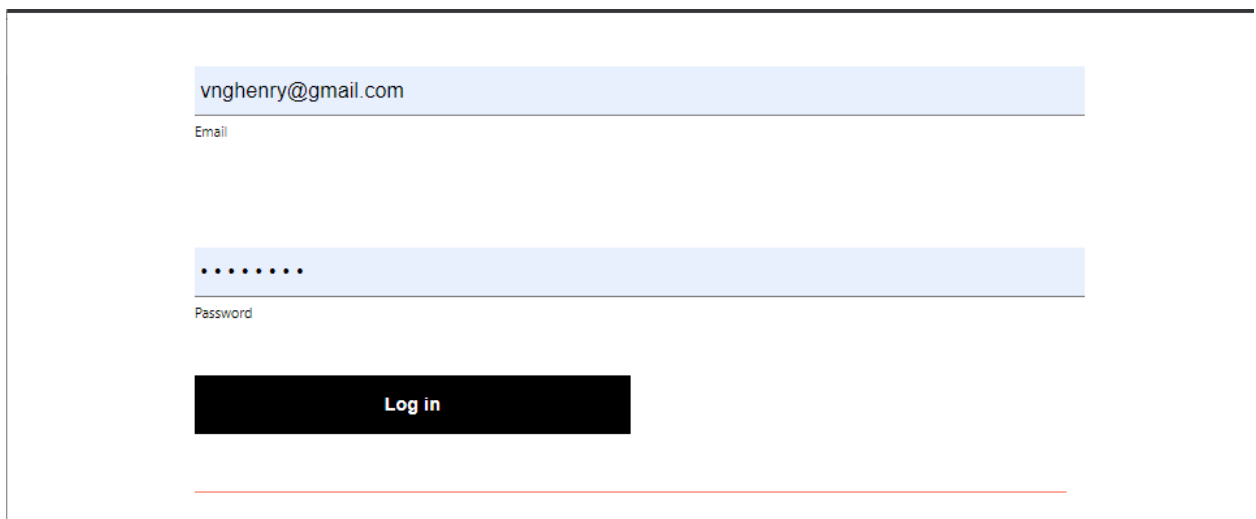
- iv. Ο χρήστης μέσω της εφαρμογής μπορεί να διαγράψει συγκεκριμένη εξέταση. Μέσω της μεθόδου DELETE και στο path localhost:3000/examinations/:id η εφαρμογή επιστρέφει ένα json με τα στοιχεία της εξέτασης που διαγράφηκε.



Παρουσίαση εφαρμογής μέσω προγράμματος περιηγητή (Browser)

Login

Ο χρήστης θα πρέπει να εισάγει το σωστό username και password προκειμένου να εισέλθει στην εφαρμογή. Σε διαφορετική περίπτωση θα του εμφανιστή Pop up μήνυμα το οποίο θα τον ενημερώνει ότι τα στοιχεία που εισάγει είναι λανθασμένα. Με την είσοδο του ο χρήστης θα «κουβαλάει» μαζί του ένα access token το οποίο αποθηκεύεται στο local storage του browser. Σε κάθε επικοινωνία Frontend και Backend θα χρησιμοποιείται το access token το οποίο δηλώνει ότι χρήστης είναι εξουσιοδοτημένος και μπορεί να δει τις πληροφορίες που του αντιστοιχούν.



The image shows a login form with the following elements:

- An email input field containing the text "vnghenry@gmail.com". Below the field is the label "Email".
- A password input field with seven dots representing masked characters. Below the field is the label "Password".
- A black button with the text "Log in" in white.
- A horizontal red line is positioned below the button.

Σελίδα εξετάσεων

Ο χρήστης μπορεί να παρακολουθήσει τις εξετάσεις και να τις επεξεργαστεί, να διαγράψει και να τις ολοκληρώσει. Το UI ενημερώνεται αυτόματα μετά από κάθε επεξεργασία του χρήστη. Επίσης στο πάνω μέρος υπάρχει και το μενού όπου ο χρήστης μπορεί να περιηγηθεί στην εφαρμογή.

The screenshot shows the application's user interface. At the top, there is a header with a logo on the left, navigation links "Οι εξετάσεις μου" and "Προσθήκη" in the center, and a user profile "Henry Vangelis" with an "Εξοδος" (Logout) button on the right. The main content area displays a grid of four exam cards. Each card contains the following information:

- Exam Title:** Εξέταση: FDSF (top-left), Εξέταση: εξέταση 1 (top-right), Εξέταση: εξέταση 1 (bottom-left), Εξέταση: FDSF (bottom-right).
- Author:** Γιατρός: ιατρος 1 (all cards).
- Status:** Ολοκληρώθηκε: οχι (all cards).
- Description:** A paragraph of Lorem Ipsum text, with some variations in the bottom-right card (e.g., "από τον 15ο αιώνα" vs "από τον 15ο αιώνα").
- Actions:** "Ολοκλήρωση" (green button) and "Διαγραφή" (red button) at the bottom of each card.

Σελίδα προσθήκης εξέτασης

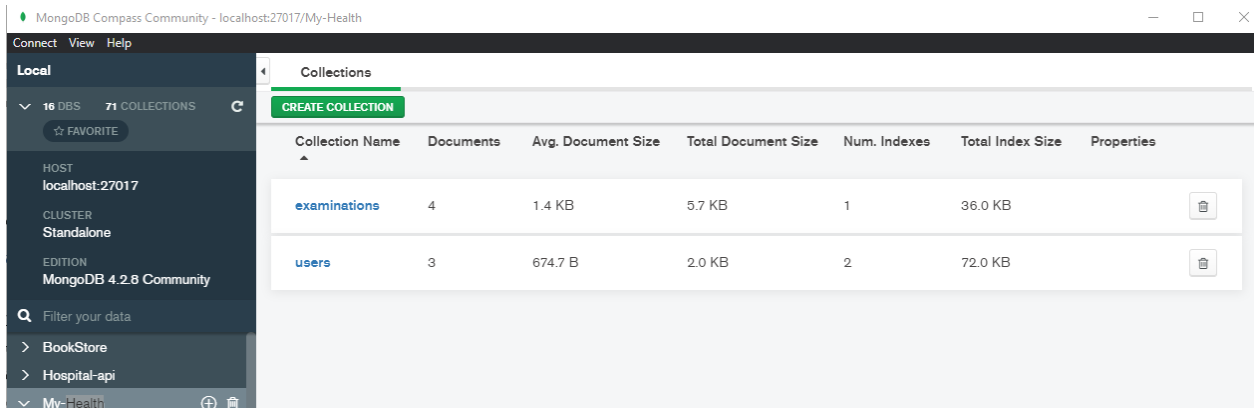
Ο χρήστης μπορεί να προσθέσει μία καινούργια εξέταση. Εφόσον συμπληρώσει όλα τα πεδία και πατήσει το κουμπί προστίθεται η εξέταση στο λογαριασμό και μπορεί να τη δει άμεσα μέσω της σελίδας οι εξετάσεις μου.

Προβολή Λογαριασμού

Ο Χρήστης μπορεί να δει τα στοιχεία του λογαριασμού του

Εξέταση

```
{
  "_id":{"$_oid":"5fc2ab3e516f241dcc7d09aa"},
  "completed":false,
  "medical_examination":"πρωτη εξεταση",
  "doctor":"ιατρος 50",
  "user":"aris vang",
  "description":"Ipsum είναι απλά ένα κείμενο χωρίς νόημα για τους επαγγελματίες της τυπογραφίας και στοιχειοθεσίας. Το Lorem Ipsum είναι το επαγγελματικό πρότυπο όσον αφορά το κείμενο χωρίς νόημα, από τον 15ο αιώνα, όταν ένας ανώνυμος τυπογράφος πήρε ένα δοκίμιο και ανακάτεψε τις λέξεις για να δημιουργήσει ένα δείγμα βιβλίου. Όχι μόνο επιβίωσε πέντε αιώνες, αλλά κυριάρχησε στην ηλεκτρονική στοιχειοθεσία, παραμένοντας με κάθε τρόπο αναλλοίωτο. Έγινε δημοφιλές τη δεκαετία του '60 με την έκδοση των δειγμάτων της Letraset όπου περιελάμβαναν αποσπάσματα του Lorem Ipsum, και πιο πρόσφατα με το λογισμικό ηλεκτρονικής σελιδοποίησης όπως το Aldus PageMaker που περιείχαν εκδοχές του Lorem Ipsum.",
  "doc_for_receipe":"εγγραφο",
  "doc_for_delivery":"dasd",
  "owner":{"$_oid":"5fc26661516f241dcc7d09a2"},
  "createdAt":{"$date":"2020-11-28T19:55:42.687Z"},"updatedAt":{"$date":"2020-11-28T20:00:53.051Z"},"__v":0}
```



MongoDB Compass Community - localhost:27017/My-Health

Connect View Help

Local

16 DBS 71 COLLECTIONS

☆ FAVORITE

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 4.2.8 Community

Filter your data

> BookStore

> Hospital-api

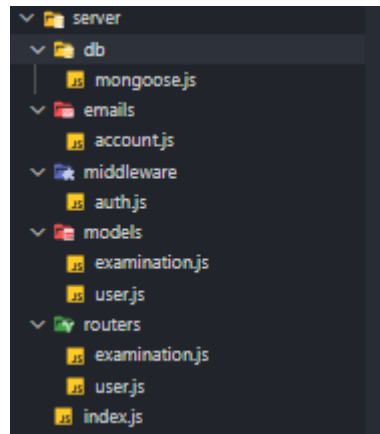
My-Health

Collections

CREATE COLLECTION

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
examinations	4	1.4 KB	5.7 KB	1	36.0 KB	
users	3	674.7 B	2.0 KB	2	72.0 KB	

Backend node JS



i. JavaScript

Είναι μία γλώσσα προγραμματισμού αντικειμενοστραφής (object oriented), η οποία είναι πολύ ελαφριά και δεν χρειάζεται μεταγλωττιστή με πρώτης κατηγορίες λειτουργίες. Η κύρια της χρήση ήταν για ανάπτυξη ιστοσελίδων δηλαδή ως γλώσσα σεναρίου σε browser. Πλέον όμως η JavaScript είναι μία δυναμική γλώσσα προγραμματισμού καθώς μπορεί να υποστηρίξει την συνολική υλοποίηση μια εφαρμογής. <https://developer.mozilla.org/el/docs/Web/JavaScript>

ii. NodeJS

Το NodeJS είναι ένα γρήγορο διαδικτυακό περιβάλλον εκτέλεσης JavaScript το οποίο χρησιμοποιείται τόσο για διακομιστές όσο και για διάφορες εφαρμογές του. Επομένως το NodeJS δίνει ταχύτητα και αποδοτικότητα στην εφαρμογή my Health. Επομένως, το NodeJS βασίζεται σε ένα runtime JavaScript περιβάλλον το οποίο επιτυγχάνει μία υψηλή απόδοση μειώνοντας την χαμηλή λανθάνουσα κατάσταση. Με άλλα λόγια δεν σπαταλάει χρόνο για input / output άλλων αιτημάτων.

Το NodeJS ακολουθεί μια διαφορετική προσέγγιση για την υλοποίησης εισερχομένων και εξερχομένων νημάτων από το διακομιστή. Πιο συγκεκριμένα εκτελεί ένα βρόχο συμβάντων με ένα συγκεκριμένο τρόπο που αποθηκεύει στο σύστημα για την αντιμετώπιση των συνδέσεων και κάθε καινούρια που εισέρχεται γίνεται επανάκλαση της JavaScript η οποία μπορεί να διαχειριστεί I/O αιτήματα με μη αποκλειστικές κλήσεις I/O και αν είναι απαραίτητο να τις αποκλείσει η να χρησιμοποιεί περισσότερο CPU αν είναι αναγκαίο. Έτσι το NodeJS απαιτεί λιγότερη μνήμη για να χειριστεί περισσότερες συνδέσεις, σε αντίθεση με άλλες τεχνολογίες όπως asp.net, ruby, java server κλπ.

iii. Express

Το express είναι το πιο γνωστό Node Web Framework. Δίνει τη δυνατότητα να δημιουργήσεις Application programming Interfaces (API) και να διαχειριστείς HTTP requests και να παράγεις responses σε διαφορετικά path. Υπεύθυνο για την ένωση των request και responses είναι το Port και το location που παράγει το response.

```
const auth = require("../middleware/auth");
const express = require("express");
const User = require("../models/user");
const router = new express.Router();
const { welcomeEmail } = require("../emails/account");
var cors = require("cors"); 5.2K (gzipped: 2.1K)

//after request option needed
router.options("/users", cors());
router.options("/users/login", cors());
router.options("/users/me", cors());
```

iv. Npm

Το NPM είναι διαχειριστής πακέτων / βιβλιοθηκών της γλώσσας προγραμματισμού JavaScript και είναι συνδεδεμένο με ένα website το οποίο λέγεται registry και περιλαμβάνει όλες τις βιβλιοθήκες ανοιχτού κώδικα του NodeJS. Εκεί υπάρχουν χιλιάδες πακέτα και modules για διάφορες χρήσεις. Το πιο δημοφιλές πακέτο είναι το React, η βιβλιοθήκη που χρησιμοποιείται στο my Health app για την ανάπτυξη της διεπαφής.

v. Sendgrid mail

Είναι mail API το οποίο σου δίνει τη δυνατότητα να στείλεις email μέσω της εφαρμογής.

```
1 const sgMail = require("@sendgrid/mail");
2
3 const sendGridAPIKey =
4   "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";
5
6 sgMail.setApiKey(sendGridAPIKey);
7
8 const welcomeEmail = (email, name) => {
9   sgMail.send({
10     to: email,
11     from: "vnghenry@gmail.com",
12     subject: "welcome to My Health",
13     text: `Hello ${name}, welcome to to My Health app`,
14   });
15 };
16
17 module.exports = {
18   welcomeEmail,
19 };
20
```

vi. Jsonwebtoken

Το JSON Web Token είναι ένας τρόπος ώστε να κρυπτογραφήσεις ένα JSON μπορεί να χρησιμοποιηθεί με το OAuth 2.0 πρωτοκολλο ώστε να κρυπτογραφήσει και να αποθηκεύσει το ίδιο το Access token στο εαυτό του αντί για τη βάση δεδομένων.

```
const jwt = require('jsonwebtoken')
const User = require('../models/user')

const auth = async (req, res, next) => {
  try {
    const token = req.header('Authorization').replace('Bearer ', '')
    const decoded = jwt.verify(token, 'hospitalAuth')
    const user = await User.findOne({
      _id: decoded._id,
      'tokens.token': token
    })

    if (!user) {
      throw new Error()
    }

    req.token = token
    req.user = user
    next()
  } catch (e) {
    res.status(401).send({
      error: 'Please Authenticate'
    })
  }
}
```

vii. Moment

Είναι μία βιβλιοθήκη η οποία σου δίνει τη δυνατότητα να διαχειριστείς την ώρα και τις ημερομηνίες μέσα στην εφαρμογή σου.

viii. Mongo dB

Η συγκεκριμένη βιβλιοθήκη είναι ο επίσημος οδηγός για το NodeJS η οποία παρέχει υψηλού επιπέδου application programming interface πάνω από mongodb-core για τους τελικούς χρήστες.

```
const express = require("express");
require("./db/mongoose"); // grab everything from this file and check if the db is connected

const userRouter = require("./routers/user");
const examinationRouter = require("./routers/examination");
const cors = require("cors"); // 5.2K (gzipped: 2.1K)

const app = express(); // getting express
const port = process.env.PORT || 3000; //heroku if does not exists go locally

const multer = require("multer");
const upload = multer({
  dest: "images",
});
app.post("/upload", upload.single("upload"), (req, res) => {
  res.send();
});

app.use(express.json()); //post man object
app.use(userRouter);
app.use(examinationRouter);
app.use(cors());

app.listen(port, () => {
  console.log("Server is up on port: " + port);
});
```

ix. Mongoose

Είναι ένα Object Document Mapper (ODM) το οποίο κάνει τη mongoDB πιο εύκολη ως προς την χρήση της καθώς μετατρέπει τα document της βάσης δεδομένων σε object της JavaScript. Ουσιαστικά το mongoose σου δίνει τη δυνατότητα να φτιάξεις μοντέλα για τη βάση δεδομένων και μέσω αυτού να κάνεις validation, type casting κλπ.

```
const userSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
    trim: true,
  },
  phone: {
    type: String,
    required: true,
    trim: true,
    minlength: 10,
    maxlength: 10,
  },
  email: {
    type: String,
    unique: true,
    required: true,
    trim: true,
    lowercase: true,
    validate(value) {
      if (!validator.isEmail(value)) {
        throw new Error('Email invalid');
      }
    },
  },
  age: {
    type: Number,
    default: 0,
    validate(value) {
      if (value < 0) {
        throw new Error('Age must be positive number');
      }
    },
  },
  password: {
    type: String,
    required: true,
    minlength: 6,
    trim: true,
    validate(value) {
      if (value.toLowerCase().includes('password')) {
        throw new Error('Password is not allowed');
      }
    },
  },
  tokens: [
    {
      token: {
        type: String,
        required: true,
      },
    },
  ],
  avatar: {
    type: Buffer,
  },
}, {
  timestamps: true,
});
```

x. Nodemon

Η βιβλιοθήκη βοηθάει στο όταν κάνεις αλλαγές στο κώδικα της εφαρμογής ο Server κάνει επανεκκίνηση αυτόματα.

xi. Multer

Το Melter δίνει τη δυνατότητα να διαχειριστής αρχεία στην εφαρμογή σου.

xii. Axios

Το axios είναι ένα promise-based HTTP client το οποίο μπορεί να χρησιμοποιεί τόσο frontend όσο και στο backend. Χρησιμοποιώντας το axios μπορείς να στείλεις asynchronous HTTP request στα REST endpoint και να πραγματοποιήσεις CRUD διαδικασίες. Επομένως, κάνει XMLHttpRequests από το διακομιστή, κάνει κλήσης από NodeJS, υποστηρίζει promise API, διαχειρίζεται κεντρικά τις απαντήσεις του server μέσω του interceptor.

```
import axios from "axios"; // 15.4k (gziped: 5.2k)

const token = localStorage.getItem("token");

const config = {
  headers: {
    Authorization: "Bearer ${token}",
  },
};

export const userProfile = async () => {
  const url = "http://localhost:3000/users/me";
  const profile = await axios.get(url, config);
  return profile;
};

export const login = async (user) => {
  const userData = await axios.post("http://localhost:3000/users/login", user);
  return userData.data.token;
};

export const getExaminations = async () => {
  const url = "http://localhost:3000/examinations?sortBy=createdAt:desc";
  const examinations = await axios.get(url, config).then((response) => {
    return response.data;
  });
  return examinations;
};

export const getProfile = async () => {
  const url = "http://localhost:3000/users/me";
  const profile = await axios.get(url, config);
  return profile;
};

export const submitExamination = async (examination) => {
  const url = "http://localhost:3000/examinations";
  const newExamination = await axios.post(url, examination, config);
  return newExamination;
};

export const updateExamination = async (completed, id) => {
  const url = "http://localhost:3000/examinations/${id}";
  const data = await axios.patch(url, { completed: completed }, config);
};

export const deleteExamination = async (id) => {
  const url = "http://localhost:3000/examinations/${id}";
  await axios.delete(url, config);
};
```

Frontend React JS

i. React

Για την ανάπτυξη του Frontend επιλέχθηκε η βιβλιοθήκη React της γλώσσας JavaScript, η οποία είναι τύπου ανοιχτού κώδικα δημιουργήθηκε και συντηρείται από την εταιρία Facebook. Ως βιβλιοθήκη δίνει τη δυνατότητα στους προγραμματιστές να την χρησιμοποιούν ώστε να δημιουργούν δυναμικά και εύκολα user interfaces μέσω των components. Θεωρείται ιδανική για σελίδες single page application η κινητών. Αυτό που κάνει δημοφιλή είναι ότι αυτοματοποιεί την παραγωγή δεδομένων στο Document Object Model.

ii. Authentication

Για την υλοποίηση του authentication χρησιμοποιήθηκε η βιβλιοθήκη JWT. Κάθε φορά που χρήστης εισέρχεται στην εφαρμογή κουβαλάει μαζί του ένα access token το οποίο χρησιμοποιείται σε κάθε επικοινωνία με το server. Επίσης ο κωδικός του χρήστη κρυπτογραφείται στη βάση δεδομένων.

iii. CSS

Για την εικαστική μορφοποίηση της σελίδα επιλέχθηκε η Cascading Style Sheet (CSS). Η CSS δίνει τη δυνατότητα να μορφοποιήσεις ιστοσελίδες οι οποίες αποτελούνται από HTML. Επίσης, αντί να γίνεται μορφοποίηση σε κάθε html item, η σχεδίαση ορίζεται σε ένα σημείο και το αντίστοιχο στοιχείο κληρονομεί τα εικαστικά στοιχεία.

Συμπεράσματα και μελλοντικές επεκτάσεις

Η εφαρμογή my Health δημιουργήθηκε με σκοπό να διευκολύνει τις διαδικασίες των εξετάσεων για τους ασθενείς, του ιατρούς και τα νοσοκομεία. Δίνει τη δυνατότητα με ένα εύκολο τρόπο ο χρήστης να διαχειριστεί τις ιατρικές του εξετάσεις έχοντας έτσι μια συνολική εικόνα για την υγεία του με το πέρασμα των χρόνων. Από την άλλη μεριά οι ιατροί εφαρμόζοντας την εφαρμογή στην εργασία τους θα μπορούν να έχουμε μια εικόνα για τους ασθενείς τους οποιαδήποτε στιγμή βοηθώντας του έτσι να είναι πιο αποδοτικοί.

Όσον αφορά τις τεχνολογίες, είναι πολύ σημαντικό ότι μέσω της κοινότητας του ανοιχτού κώδικα μπορείς δημιουργήσεις σύγχρονες και αξιόπιστες εφαρμογές οι οποίες μπορούν να εξυπηρετήσουν και διευκολύνουν τις διαδικασίες του απλού κόσμου.

Στη εφαρμογή my Health θα μπορούσαν να γίνουν πολλές επεκτάσεις καθώς ο χώρος τον οποίο εξυπηρετεί είναι τεράστιος και με πολλές ανάγκες.

Πρώτη επέκταση θα είναι η δυνατότητα ο χρήστης να μπορεί να κλείνει ραντεβού με τον ιατρό του οποιαδήποτε στιγμή μέσω της εφαρμογής.

Δεύτερη, οι πληρωμές να γίνονται μέσω της εφαρμογής.

Τρίτη, με βάση τις εξετάσεις θα δημιουργηθεί αλγόριθμος ο οποίος θα βγάζει στατιστικά για την υγεία του ασθενή, προκειμένου να κάνει προβλέψεις και δίνει προτάσεις στον ίδιο τον ασθενή.

Τέταρτη, μετά της εξέταση ο ιατρός θα μπορεί να προσθέτει τα αντίστοιχα φάρμακα έτσι ο χρήστης να τα παραγγείλει άμεσα.

Πέμπτη, στόχος είναι στην εφαρμογή να συμμετέχουν και τα νοσοκομεία προκειμένου να αυτοματοποιηθεί η διαδικασία εισαγωγής του ασθενή στο νοσοκομείο. Πχ όταν ο ασθενής επισκεφθεί το νοσοκομείο και είναι σε θέση να μεταβεί χωρίς βοήθεια στα ιατρεία θα του προστίθενται οι εξετάσεις που θα πρέπει να κάνει και στη συνέχεια τα αποτελέσματα τους χωρίς να χρειάζεται να περιμένει.

Βιβλιογραφία

Mozilla, 2020 MDN contributors, αναρτήθηκε 23 Νοέμβριου 2013 από

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction

INNOVIEW (2018). 2018 τι είναι το node JS, αναρτήθηκε 27 Απριλίου 2018

<https://www.innoview.gr/el/blog/general/545-ti-einai-to-node-js>

LINUX INSIDER (2019). 2019 Τι είναι το npm και πως το χρησιμοποιούμε για εγκατάσταση-δημιουργία Node.js πακέτων, αναρτήθηκε 17 Μαρτίου 2019 από

<https://www.linuxinsider.gr/forum/programmatismos/9585/prota-bimata-sto-web-development-ti-einai-npm-kai-pos-hrisimopoioume-gia>

NPM (2020), The official MongoDB driver for Node.js αναρτήθηκε 11 Νοεμβρίου 2020 από

<https://www.npmjs.com/package/mongodb>

oauth(χ,χ). JSON Web Token, αναρτήθηκε 12 Δεκέμβριου 2020 από

<https://oauth.net/2/jwt/>

REACT(χ,χ) documentation, αναρτήθηκε 5 Δεκεμβριου 2020 απο

<https://reactjs.org/docs>

Red Hat (χ.χ.) What is an API, ανακτήθηκε 21 Σεπτεμβρίου 2020 από

<https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>.

Microsoft (2018). Web API design, ανακτήθηκε 21 Σεπτεμβρίου 2020 από <https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design..>

.